Esercizio "Il vostro turno" con display LCD; la pratica

***L**'ultimo montaggio che realizzeremo sul laboratorio per provare uno degli esercizi presentati corrisponde all'applicazione "Il vostro turno" utilizzando il modulo LCD. Questo montaggio, come la maggior parte di quelli in cui si utilizza il display LCD, è semplice, e risulterà molto facile da implementare.*

Scrittura del microcontroller

Compilando il codice "turno.asm" (esercizio: Il vostro turno su LCD), generiamo il file in codice macchina "turno.hex", ed effettuata la simulazione in modo soddisfacente lo dobbiamo caricare sul PIC. Questa procedura dipende da due fattori, hardware e software. quindi li analizzeremo e li configureremo per poter eseguire questo processo.

Configurazione del laboratorio per la scrittura

È necessario ripetere i passaggi da eseguire per configurare correttamente il laboratorio per la scrittura, dato che ripetendoli ci risulterà più facile farli diventare un gesto meccanico, quasi un'abitudine che applicheremo a tutti gli esercizi. Il microcontroller deve essere inserito sullo zoccolo nella posizione corretta e i jumper della scheda DG06, JP1, JP2 e JP3, devono avere i ponticelli sulle posizioni 1 e 2. I jumper JP8 e JP9 devono essere inseriti per poter scrivere il PIC, infine è necessario collegare il cavo di scrittura fra il laboratorio e il PC. Il laboratorio dovrà avere l'aspetto che vediamo nell'immagine della pagina successiva.

Software IC-Prog

Configurato l'hardware in modo che accetti la scrittura, non ci rimane che effettuarla, quindi apriamo il software IC-Prog e seguiamo i passaggi indicati per questo processo. Eseguiremo la cancellazione del dispositivo selezionando l'opzione Cancella tutto. Per verificare che la cancellazione venga eseguita correttamente leggeremo il dispositivo e controlleremo che in tutti gli indirizzi di memoria appaia il valore 3FFF.

Selezioniamo il programma che vogliamo scrivere sul PIC, apriamo il file "turno.hex" e verifichiamo come cambia la videata che contiene il codice di programma. Prima di scrivere verifichiamo che il dispositivo selezionato sia il

File risultanti dalla compilazione.

Configurazione del laboratorio per la scrittura del PIC.

Colleghiamo la Porta B (uscite) con il display LCD (ingressi).

PIC 16F870 e che la protezione del codice si trovi su CP OFF. Configuriamo l'oscillatore selezionando XT, e la parola di configurazione selezionando i bit WDT e PWRT. Fatto questo possiamo selezionare l'opzione Programma tutto per caricare il programma selezionato sul PIC. L'ultimo passaggio consiste nel leggere nuovamente il dispositivo per verificare che il processo di scrittura sia stato eseguito con successo.

Montaggio dell'applicazione

Abbiamo ora il PIC debitamente caricato con il programma con cui vogliamo lavorare sul laboratorio, ma prima di eseguire il montaggio necessario per far girare l'applicazione è necessario configurare nuovamente il laboratorio per il modo lavoro.

Scolleghiamo il cavo di trasferimento fra il

Carichiamo su IC-Prog il programma "turno.hex".

Montaggio dei segnali E, RS e R/W.

Montaggio degli ingressi sulla scheda Bread Board.

PC e il laboratorio, togliamo i ponticelli dei connettori JP8 e JP9 e spostiamo di posizione i ponticelli dei connettori JP1, JP2 e JP3.

Montaggio delle uscite

Per eseguire il montaggio delle uscite dobbiamo solamente unire mediante due cavetti a quattro fili i connettori corrispondenti alla porta B con i connettori corrispondenti alle linee di ingresso dei dati del modulo LCD (quello centrale e quello di destra del pannello superiore del laboratorio). L'unico dettaglio a cui è necessario prestare un po' più di attenzione è mantenere la corrispondenza fra i pin della porta con quelli del display LCD, in altre parole non dobbiamo incrociare i cavi.

Montaggio degli ingressi

In tutti i montaggi che comprendono un display LCD sono necessari tre ingressi riservati a esso: E, RS, R/W, quindi li dovremo prevedere sul laboratorio. Questi segnali saranno forniti dal PIC tramite le sue uscite RA2:RA0, quindi collegheremo mediante un connettore a quattro fili i pin RA3:RA0 della porta A con le molle 5, 6 e 7 (il filo corrispondente a RA3 rimarrà libero). Mediante un altro cavetto a quattro fili terminato su due connettori, uniremo il connettore che corrisponde alle molle azzurre con il connettore del display LCD situato a sinistra della fila dei connettori (l'unico libero, dato che gli altri due sono stati collegati alla porta B). Ricordate che dovrete mantenere la corrispondenza fra i pin della porta A e quelli del display LCD.

Nell'applicazione che vogliamo provare è sufficiente il montaggio di un ingresso che simuli il pulsante della macchina "Il vostro turno". Inseriamo una resistenza sulla scheda Bread Board che da un lato sia collegata a 5 V e dall'altro alla molla di interconnessione numero 15. La molla 16 la uniremo direttamente a 0 V. Mediante un cavetto a due fili collegheremo il connettore corrispondente alle molle verdi (15 e 16) con il pulsante P2. Infine mediante un filo, collegheremo la molla 15 con il

Colleghiamo il pulsante al circuito di ingresso.

Montaggio finale dell'applicazione.

Prova di funzionamento.

pin di ingresso RA4, per far arrivare al PIC l'ingresso del pulsante P2.

Prova di funzionamento

Terminato il montaggio dell'applicazione dobbiamo alimentare il laboratorio e verificare il funzionamento. Se sul display abbiamo regolato bene il contrasto potremo vedere che si visualizza il messaggio "TURNO:00". Fino a quando non attiveremo l'ingresso si manterrà il messaggio sul display.

Se premiamo il pulsante P2, quando questo ritorna nella sua posizione originale, sul display si visualizzerà il seguente numero "TURNO:01". Ogni volta che premiamo sull'ingresso si incrementa il contatore, quindi la macchina che abbiamo sviluppato funziona correttamente.

Raccomandazioni

Un'applicazione non deve essere considerata terminata fino a quando non vengono provate tutte le possibili combinazioni o alternative. Nell'esercizio di cui ci stiamo occupando, non abbiamo verificato che cosa succede quando il contatore arriva a 99, né cosa succede quando manca l'alimentazione al laboratorio. Vagliate tutte le combinazioni prima di dichiarare terminato un lavoro.

Continuate a fare esercizi con il laboratorio, risolvendo nuovi progetti o modificando quelli già esistenti. Per esercitarvi con questa applicazione potete fare delle modifiche nel programma, come ad esempio utilizzare la seconda fila di caratteri dell'LCD, per visualizzare i cicli eseguiti dal contatore, oppure contare di due in due, caricare il programma della scheda Smart Card o qualsiasi altra cosa vi venga in mente.

Presentazione sul display LCD.

ERRATA CORRIGE

Scheda: Digitale avanzato 1
Porte, livelli e ritardi

Il numero della scheda è sbagliato, non 1 ma 5

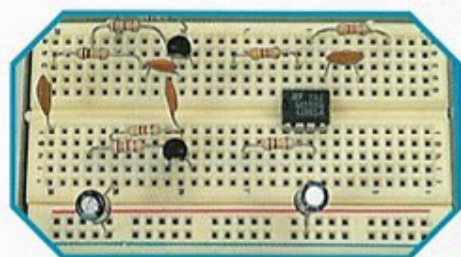
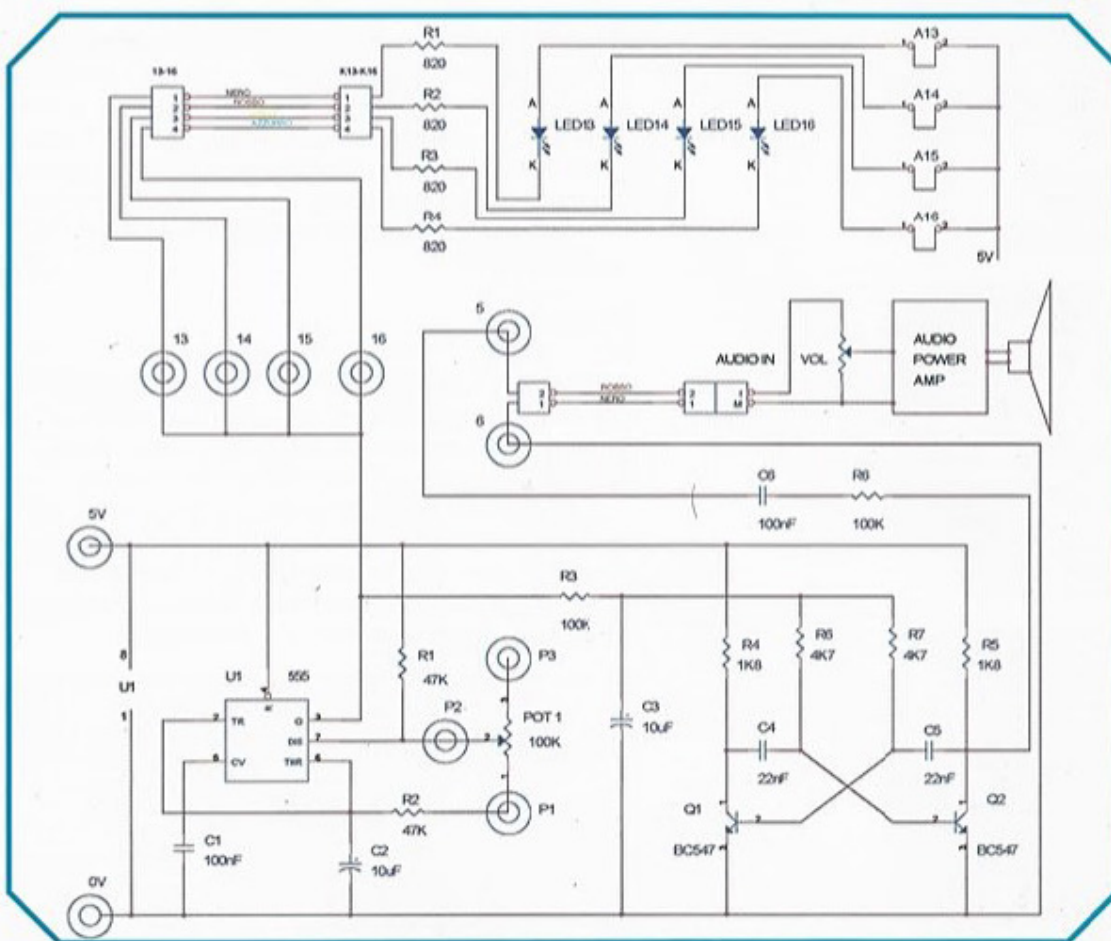
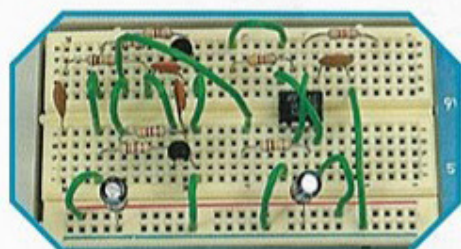
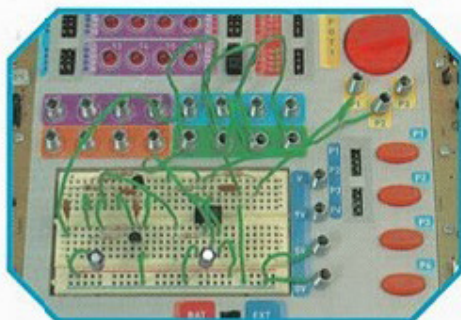
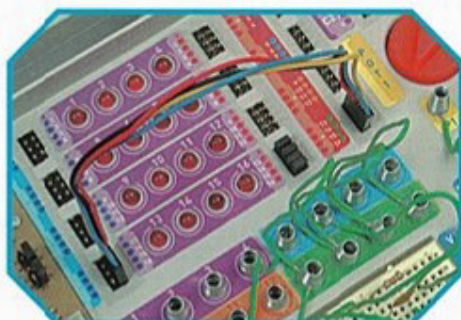
Generatore audio modulato

Questo circuito corrisponde a un VCO (Oscillatore Controllato in Voltaggio), la cui tensione di controllo è generata da un altro oscillatore.

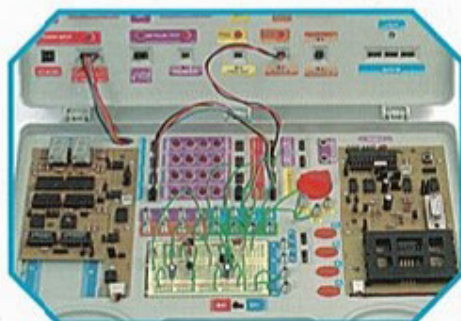
Il circuito

Osservando il circuito possiamo vedere un VCO formato principalmente dai transistor Q1 e Q2 e dai rispettivi componenti passivi associati. L'ingresso di controllo della frequenza di questo oscillatore è il punto di collegamento delle resistenze R6 e R7.

Nello schema possiamo anche vedere un circuito integrato 555 configurato per lavorare come oscillatore astabile, ma a una frequenza molto bassa. La sua uscita si può utilizzare per controllare il VCO e si collega a esso tramite una rete RC. Per fare in modo che la variazione di tensione non sia brusca, la rete RC è formata dalla resistenza R3 e dal condensatore C3.

Distribuzione dei componenti.

Collegamenti della scheda.

Schema dell'oscillatore audio modulato.

Dettaglio del collegamento alle molle.

Dettaglio del collegamento ai LED.

Collegamenti all'amplificatore audio.

Ruotando POT 1 cambia la modulazione.

Esperimento completato.

La tensione di uscita terminale 3 del 555, si utilizza per illuminare la quarta fila di LED, cosa che avviene solamente quando questo livello di uscita è basso. L'uscita del VCO è portata verso l'amplificatore audio inserendo un circuito seriale formato dalla resistenza R6 e dal condensatore C6, in modo che ascoltando i suoni sull'altoparlante si possa verificare l'effetto delle variazioni di livello sull'uscita del 555.

Montaggio

In questo montaggio è necessario inserire diversi componenti sulla scheda Bread Board, quindi dovremo essere ordinati ed eseguire il lavoro lentamente per evitare errori.

I collegamenti dei catodi dei LED dal 13 al 16, si eseguono con un cavetto terminato su due connettori a quattro vie cadauno, che arriva al connettore corrispondente alle molle dalla 13 alla 16, inoltre dovremo inserire i quattro ponticelli sugli anodi di questi LED.

I collegamenti dell'uscita del VCO all'amplificatore audio si eseguono con un cavetto terminato su due connettori a due vie. Uno di questi connettori si collega a quello corrispondente alle molle 5 e 6 e l'altro capo al connettore AUDIO IN, in modo che il filo rosso corrisponda al punto rosso e quello nero al punto nero. Ricordate che l'uscita dell'amplificatore per essere collegata deve avere i due ponticelli inseriti in senso orizzontale, fra AUDIO OUT e SPEAKER IN.

Prova

Dopo aver verificato il montaggio possiamo dare alimentazione al circuito, includendo in questa operazione il commutatore AUDIO ON, in modo che funzioni l'amplificatore.

Dopo aver messo in marcia il circuito si udirà un suono sull'altoparlante, la cui frequenza si potrà modulare azionando il comando di POT 1. Ogni volta che cambia il suono i LED dal 13 al 16 si spengono o si accendono.

LISTA DEI COMPONENTI

U1	Circuito integrato 555
Q1, Q2	Transistor BC547 o BC548
R1, R2	Resistenza 47 K (giallo, viola, arancio)
R3, R6	Resistenza 100 K (marrone, nero, giallo)
R4, R5	Resistenza 1K8 (marrone, grigio, rosso)
R6, R7	Resistenza 4K7 (giallo, viola, rosso)
C1, C6	Condensatore 22 nF
C2, C3	Condensatore 10 µF elettrolitico
C4, C5	Condensatore 22 nF

Esercizio "Il vostro turno" con display LCD; la pratica

***L**'ultimo montaggio che realizzeremo sul laboratorio per provare uno degli esercizi presentati corrisponde all'applicazione "Il vostro turno" utilizzando il modulo LCD. Questo montaggio, come la maggior parte di quelli in cui si utilizza il display LCD, è semplice, e risulterà molto facile da implementare.*

Scrittura del microcontroller

Compilando il codice "turno.asm" (esercizio: Il vostro turno su LCD), generiamo il file in codice macchina "turno.hex", ed effettuata la simulazione in modo soddisfacente lo dobbiamo caricare sul PIC. Questa procedura dipende da due fattori, hardware e software. quindi li analizzeremo e li configureremo per poter eseguire questo processo.

Configurazione del laboratorio per la scrittura

È necessario ripetere i passaggi da eseguire per configurare correttamente il laboratorio per la scrittura, dato che ripetendoli ci risulterà più facile farli diventare un gesto meccanico, quasi un'abitudine che applicheremo a tutti gli esercizi. Il microcontroller deve essere inserito sullo zoccolo nella posizione corretta e i jumper della scheda DG06, JP1, JP2 e JP3, devono avere i ponticelli sulle posizioni 1 e 2. I jumper JP8 e JP9 devono essere inseriti per poter scrivere il PIC, infine è necessario collegare il cavo di scrittura fra il laboratorio e il PC. Il laboratorio dovrà avere l'aspetto che vediamo nell'immagine della pagina successiva.

Software IC-Prog

Configurato l'hardware in modo che accetti la scrittura, non ci rimane che effettuarla, quindi apriamo il software IC-Prog e seguiamo i passaggi indicati per questo processo. Eseguiremo la cancellazione del dispositivo selezionando l'opzione Cancella tutto. Per verificare che la cancellazione venga eseguita correttamente leggeremo il dispositivo e controlleremo che in tutti gli indirizzi di memoria appaia il valore 3FFF.

Selezioniamo il programma che vogliamo scrivere sul PIC, apriamo il file "turno.hex" e verifichiamo come cambia la videata che contiene il codice di programma. Prima di scrivere verifichiamo che il dispositivo selezionato sia il

File risultanti dalla compilazione.

Configurazione del laboratorio per la scrittura del PIC.

Colleghiamo la Porta B (uscite) con il display LCD (ingressi).

PIC 16F870 e che la protezione del codice si trovi su CP OFF. Configuriamo l'oscillatore selezionando XT, e la parola di configurazione selezionando i bit WDT e PWRT. Fatto questo possiamo selezionare l'opzione Programma tutto per caricare il programma selezionato sul PIC. L'ultimo passaggio consiste nel leggere nuovamente il dispositivo per verificare che il processo di scrittura sia stato eseguito con successo.

Montaggio dell'applicazione

Abbiamo ora il PIC debitamente caricato con il programma con cui vogliamo lavorare sul laboratorio, ma prima di eseguire il montaggio necessario per far girare l'applicazione è necessario configurare nuovamente il laboratorio per il modo lavoro.

Scolleghiamo il cavo di trasferimento fra il

Carichiamo su IC-Prog il programma "turno.hex".

Montaggio dei segnali E, RS e R/W.

Montaggio degli ingressi sulla scheda Bread Board.

PC e il laboratorio, togliamo i ponticelli dei connettori JP8 e JP9 e spostiamo di posizione i ponticelli dei connettori JP1, JP2 e JP3.

Montaggio delle uscite

Per eseguire il montaggio delle uscite dobbiamo solamente unire mediante due cavetti a quattro fili i connettori corrispondenti alla porta B con i connettori corrispondenti alle linee di ingresso dei dati del modulo LCD (quello centrale e quello di destra del pannello superiore del laboratorio). L'unico dettaglio a cui è necessario prestare un po' più di attenzione è mantenere la corrispondenza fra i pin della porta con quelli del display LCD, in altre parole non dobbiamo incrociare i cavi.

Montaggio degli ingressi

In tutti i montaggi che comprendono un display LCD sono necessari tre ingressi riservati a esso: E, RS, R/W, quindi li dovremo prevedere sul laboratorio. Questi segnali saranno forniti dal PIC tramite le sue uscite RA2:RA0, quindi collegheremo mediante un connettore a quattro fili i pin RA3:RA0 della porta A con le molle 5, 6 e 7 (il filo corrispondente a RA3 rimarrà libero). Mediante un altro cavetto a quattro fili terminato su due connettori, uniremo il connettore che corrisponde alle molle azzurre con il connettore del display LCD situato a sinistra della fila dei connettori (l'unico libero, dato che gli altri due sono stati collegati alla porta B). Ricordate che dovrete mantenere la corrispondenza fra i pin della porta A e quelli del display LCD.

Nell'applicazione che vogliamo provare è sufficiente il montaggio di un ingresso che simuli il pulsante della macchina "Il vostro turno". Inseriamo una resistenza sulla scheda Bread Board che da un lato sia collegata a 5 V e dall'altro alla molla di interconnessione numero 15. La molla 16 la uniremo direttamente a 0 V. Mediante un cavetto a due fili collegheremo il connettore corrispondente alle molle verdi (15 e 16) con il pulsante P2. Infine mediante un filo, collegheremo la molla 15 con il

Colleghiamo il pulsante al circuito di ingresso.

Montaggio finale dell'applicazione.

Prova di funzionamento.

pin di ingresso RA4, per far arrivare al PIC l'ingresso del pulsante P2.

Prova di funzionamento

Terminato il montaggio dell'applicazione dobbiamo alimentare il laboratorio e verificare il funzionamento. Se sul display abbiamo regolato bene il contrasto potremo vedere che si visualizza il messaggio "TURNO:00". Fino a quando non attiveremo l'ingresso si manterrà il messaggio sul display.

Se premiamo il pulsante P2, quando questo ritorna nella sua posizione originale, sul display si visualizzerà il seguente numero "TURNO:01". Ogni volta che premiamo sull'ingresso si incrementa il contatore, quindi la macchina che abbiamo sviluppato funziona correttamente.

Raccomandazioni

Un'applicazione non deve essere considerata terminata fino a quando non vengono provate tutte le possibili combinazioni o alternative. Nell'esercizio di cui ci stiamo occupando, non abbiamo verificato che cosa succede quando il contatore arriva a 99, né cosa succede quando manca l'alimentazione al laboratorio. Vagliate tutte le combinazioni prima di dichiarare terminato un lavoro.

Continuate a fare esercizi con il laboratorio, risolvendo nuovi progetti o modificando quelli già esistenti. Per esercitarvi con questa applicazione potete fare delle modifiche nel programma, come ad esempio utilizzare la seconda fila di caratteri dell'LCD, per visualizzare i cicli eseguiti dal contatore, oppure contare di due in due, caricare il programma della scheda Smart Card o qualsiasi altra cosa vi venga in mente.

Presentazione sul display LCD.

ERRATA CORRIGE

Scheda: Digitale avanzato 1
Porte, livelli e ritardi

Il numero della scheda è sbagliato, non 1 ma 5

Esercizio "Il vostro turno" con display LCD, compilazione e simulazione

Terminiamo lo sviluppo del codice per risolvere l'esercizio della macchina "Il vostro turno" utilizzando il display LCD eseguendo la sua compilazione e simulazione, come fasi precedenti al montaggio elettrico dell'applicazione.

Programma principale

Continueremo vedendo come adattare il codice del programma "turno.asm" (esercizio: La memoria EEPROM dei dati. La macchina "IL VOSTRO TURNO") alle nuove specifiche con il file LCD, questo file si trova nella cartella varie. Abbiamo già visto come cambia la configurazione dei dispositivi e che adesso sono due i dati da leggere nella memoria, decine e unità, con le conseguenti verifiche dei valori all'interno dei limiti.

A partire dall'etichetta Loop visualizzeremo i dati contenuti nelle nostre variabili. Prima avevamo solamente una variabile, ricercavamo la sua equivalenza nei 7 segmenti e fornivamo un'uscita. Ora dobbiamo posizionare il cursore del display LCD, e visualizzare una ad una le variabili dopo averle convertite in codice ASCII. Nell'immagine della figura possiamo vedere il modo per risolvere ciò che abbiamo spiegato.

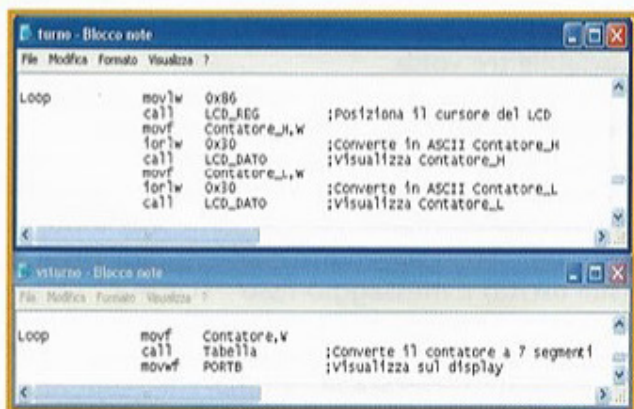
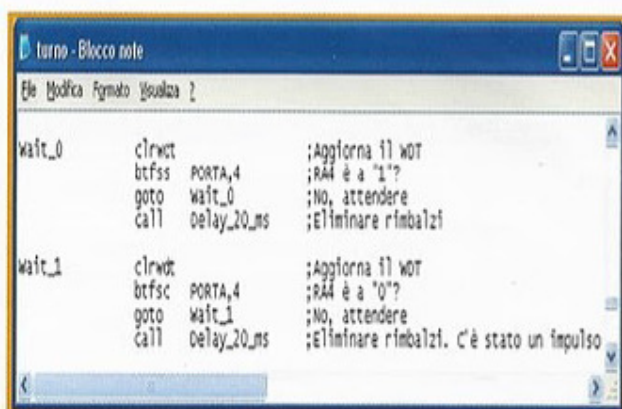
Il passo successivo è attendere che si generi un evento sull'ingresso RA4, questa parte viene risolta esattamente come nel programma base. Se RA4 è 0 attenderemo fino a quando diventa 1, e quando ritornerà nuovamente a 0, classificheremo l'impulso come valido.

Se sull'ingresso è stato generato un segnale valido dobbiamo incrementare il valore delle unità, verificando che il valore non superi il limite, cioè che non sia maggiore di 9. Se supera 9 lo riportiamo a 0 e incrementiamo le decine, verificando che anche queste non siano maggiori di 9.

Nel caso in cui anche questo valore superi il 9, il contatore avrebbe il valore di 99, quindi prima di fare la somma dovremmo resettare e il conteggio inizierebbe nuovamente da 0.

Sia nel caso precedente, che nel caso in cui il valore dopo essere stato incrementato sia inferiore o uguale a 99, lo dobbiamo scrivere nella memoria EEPROM del PIC. Per fare questo dobbiamo scrivere le due variabili, Contatore_H e Contatore_L, in indirizzi successivi nella memoria. Terminiamo il codice mediante la direttiva END. Anche se abbiamo utilizzato il programma "turno.asm" (esercizio: La memoria EEPROM dei dati. La macchina "IL VOSTRO TURNO" nella cartella varie) come base, il programma risultante è piuttosto diverso dall'originale. Salviamo il nostro codice con un nome diverso, o in una posizione diversa e verifichiamo che non abbia errori.

Compilazione

Di solito è più facile commettere errori quando si modifica un programma che quando lo si realizza dall'inizio. È piuttosto comune dimen-

Visualizzazione del dato contenuto nelle variabili.

Verifica del valore dell'ingresso.

turno - Blocco note

File Modifica Formato Visualizza ?

```
        incf    Contatore_L,F    ;Incrementa Contatore_L
        movlw   .10
        subwf   Contatore_L,W
        btfss   STATUS,Z         ;Contatore maggiore di 9?
        goto    Write            ;No, aggiorna EEPROM
        clrf    Contatore_L      ;Sì, Contatore_L passa a 0
        incf    Contatore_H,F    ;Incrementa Contatore_H
        movlw   .10
        subwf   Contatore_H,W
        btfss   STATUS,Z         ;Contatore_H maggiore di 9?
        goto    Write            ;No, aggiorna EEPROM
        clrf    Contatore_H      ;Sì, Contatore_H passa a 0
```

È stato generato un impulso, incrementiamo quindi il contatore.

ticare qualche cambio di banco di memoria per riferirci a dei registri, o lasciare qualche istruzione che invece avremmo dovuto cancellare dal programma. Per compilare questa applicazione apriamo MPLAB e creiamo un nuovo progetto, ad esempio "turno.pjt". A questo progetto allegheremo il nostro codice nella finestra di edizione del progetto, selezionando l'opzione Add Node.

Eseguendo la compilazione il processo si esegue correttamente e nella finestra dei risultati troviamo 21 messaggi di posizione dei registri sui banchi di memoria. Il programma è stato compilato senza errori e viene generato il file in codice macchina necessario per scrivere il PIC.

Simulazione

Per simulare questa applicazione dobbiamo fare praticamente la stessa cosa già fatta per gli.

turno - Blocco note

File Modifica Formato Visualizza ?

```
Write       bsf     STATUS,RP1       ;Seleziona banco 2
            clrf    EEADR            ;Seleziona indirizzo 00 della EEPROM
            bcf     STATUS,RP1       ;Seleziona banco 0
            movf    Contatore_H,W
            bsf     STATUS,RP1       ;Seleziona banco 2
            movwf   EEDATA
            call    EE_Write         ;Scrive Contatore_H nella EEPROM
            bsf     STATUS,RP1       ;Seleziona banco 2
            incf    EEADR,F          ;Indirizzo 01 della EEPROM
            bcf     STATUS,RP1       ;Seleziona banco 0
            movf    Contatore_L,W
            bsf     STATUS,RP1       ;Seleziona banco 2
            movwf   EEDATA
            call    EE_Write         ;Scrivi Contatore_L nella EEPROM
            goto    Loop


            end                      ;Fine del programma sorgente
```

Scriviamo i valori nella memoria.

esercizi precedenti. Apriamo la solita finestra dei Registri delle Funzioni Speciali e la finestra specifica per i registri più importanti. In quest'ultima inseriremo la Porta A in formato binario, la Porta B in formato ASCII, le variabili Contatore_H e Contatore_L anch'esse in ASCII, infine le variabili Temporale_1 e Temporale_2, la prima in binario e la seconda in ASCII. Quando inizieremo a simulare il programma vedremo che viene eseguito correttamente fino a quando arriva alla chiamata della routine LCD_INI. A questo punto il programma esegue le istruzioni della libreria del display LCD e quando entra nella routine LCD_DELAY temporizza 5 secondi. Per uscire da questa temporizzazione dobbiamo utilizzare la finestra Modify e forzare i valori delle variabili Lcd_Temp_1 e Lcd_Temp_2 a 1. Ricordate che la sequenza di inizializzazione implica di entrare tre volte nella stessa routine, quindi dovremo forzare il valore di ogni variabile tre volte.

Nella figura alla pagina seguente possiamo vedere i campi della finestra Modify, applicati a una delle variabili per uscire da questa routine di temporizzazione. Se continuiamo con la simulazione possiamo vedere come mediante la routine "Messaggi" presentiamo sul display (sull'uscita) il messaggio fisso "TURNO:" carattere per carattere. In seguito si legge nella memoria il valore scritto e dato che è maggiore di 9, vengono resettate le variabili che contengono le decine e le unità, cioè si inizia il conteggio. Il valore '00' viene visualizzato sul display

```
Build Results
Compiling TURNO.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 36 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 37 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 39 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 41 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 42 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 56 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 57 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 110 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 112 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 114 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 116 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 127 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 131 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 141 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 145 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 149 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 154 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 191 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 195 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 198 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TURNO.ASM 202 : Register in operand not in bank 0.  Ensure

Build completed successfully.
```

Risultato della compilazione.

Forzando l'ingresso il programma salta alla routine di eliminazione dei rimbalzi, da cui potremo uscire modificando il valore del bit T0IF a 1 (utilizzate la finestra Modify). Fatto questo, clicchiamo un'altra volta il pulsante di RA4 del simulatore in modo che torni a 0 e il programma capirà che è stato generato un ingresso valido.

Usciamo nuovamente dalla routine di eliminazione dei rimbalzi e completiamo la simulazione verificando l'incremento del contatore e che sulla porta di uscita vengano mandati i dati 0 e 1 in formato ASCII.

Possiamo continuare con la simulazione e vedere come si incrementa il contatore, anche se a questo punto possiamo considerare la simulazione come corretta.

LCD o nel caso della simulazione, sulla porta di uscita.

Il programma si ferma nell'attesa che venga generato un ingresso, che porti un 1 su RA4, quindi forzeremo questo valore con il Simulatore di Stimoli Asincroni. Assegniamo un pulsante del simulatore al terminale RA4, come possiamo vedere nella figura della pagina successiva.

Assegniamo l'opzione 'Toggle' al pulsante, per fare in modo che ogni volta che lo attiveremo faccia cambiare di stato l'ingresso.

Conclusioni

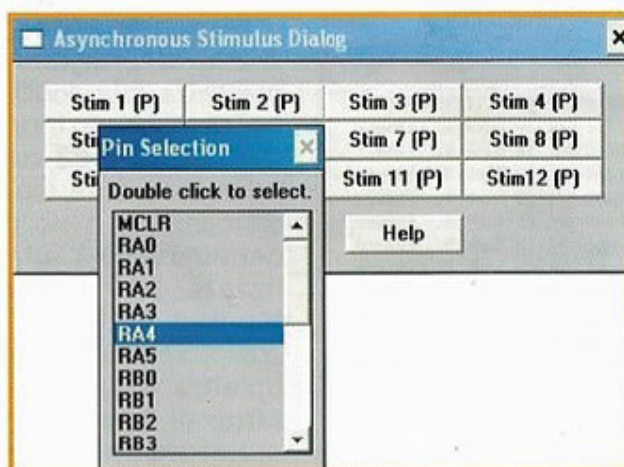
Con questo esercizio terminiamo la sezione dei microcontroller. La teoria del PIC16F870 ci

Watch_1

Address	Symbol	Value
05	PORTA	B'00000000'
06	PORTB	A'8'
22	Contatore_H	A'.'
21	Contatore_L	A'.'
23	Temporale_1	B'00000000'
24	Temporale_2	A'.'

Finestra con i registri più importanti.

Modify
Address: Lcd_Temp_2
End Address:
Data/Opcode: 1
Fill Range
Read Range
Radix: Hex Decimal
Auto Increment
Memory Area: Data Program Stack EEPROM
Write Read Close Help

Finestra Modify.

Simulatore di stimoli asincroni.

è servita per prendere contatto e capire questo piccolo dispositivo che realizza molte cose. Gli esercizi ci sono serviti per usare i principali dispositivi del microcontroller e come lavorare con le diverse periferiche. Speriamo che continuiate a praticare e a divertirvi con il mondo dei microcontroller e che otteniate il massimo rendimento dal vostro laboratorio di Elettronica Digitale.

ERRATA CORRIGE

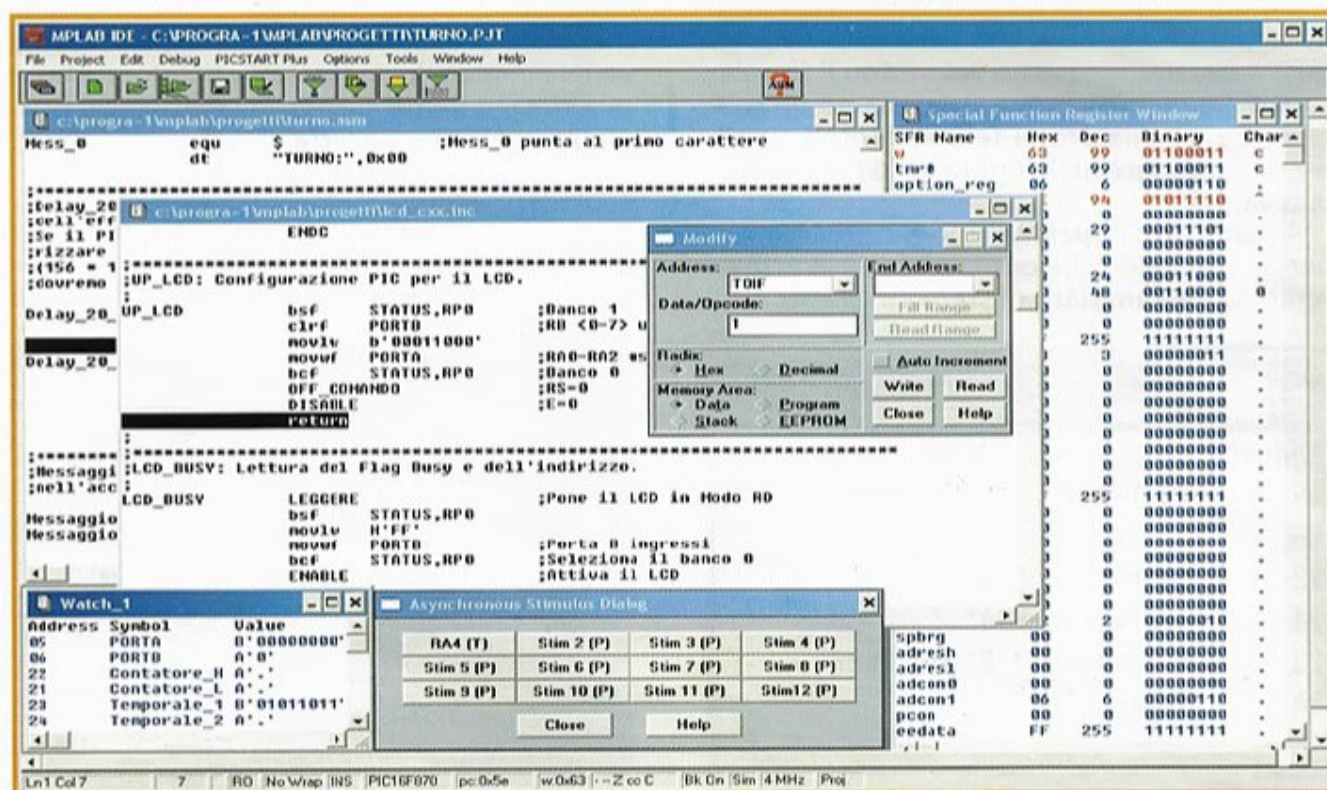
Scheda: Microcontroller 48 Il repertorio delle istruzioni

Esempio 2 di utilizzo delle istruzioni aritmetiche

Calcolare il valore con il quale rimane caricato il registro W dopo aver eseguito il seguente programma:

```
MOVLW       B7h
SWAPF       W,0
DECF        W,0
SUBLW       FEh
SOLUZIONE:  W = 1000 0100
```

Scheda: Microcontroller 51 Il repertorio delle istruzioni

Dovrebbe essere:
Il repertorio delle istruzioni (III)

MPLAB durante la simulazione.